



ФАЙЛОВЫЕ ТИПЫ ЯЗЫКА ВЫСОКОГО УРОВНЯ PASCAL

*Память человека есть лист белой бумаги:
иногда напишется хорошо, а иногда дурно.*

К. Прутков

Содержание

2

- Классификация файлов
- Основные операции с файлами
- Примеры

Классификация файлов

3

- *Файл* – именованная совокупность взаимосвязанных данных, хранящихся на внешних носителях. Файлы состоят из компонент, называемых *записями*.
- Виды файлов
 - ▣ Типизированные файлы
 - ▣ Нетипизированные файлы
 - ▣ Текстовые файлы

Типизированные файлы

4

□ *Типизированный файл*

- Последовательность значений-записей одинакового типа во внутреннем (двоичном) представлении.
- Все записи типизированного файла имеют одинаковую длину, определяемую типом значений.
- **<Имя переменной>: file of <Тип записей>**
<Тип записей> – любой, кроме файлового.

Пример: файл записей

5

```
type
  Person = record
    Name: String[10];
    Gender: Char;
    Age: Byte;
end;
```

```
var
  PersonFile:
    file of Person;
```

0-я запись													1-я запись									
1	2	3	4	5	6	7	8	9	10	11	12	13	1	2	3	4	5	6	7	8	9	10
#6	И	в	а	н	о	в	#0	#0	#0	#0	М	#20	#8	С	и	д	о	р	о	в	а	#0

Нетипизированные файлы

6

- *Нетипизированный файл*
 - Последовательность записей фиксированной длины.
 - В файл может быть записано значение любого типа, имеющее указанную длину.
 - <Имя переменной>: **file**

Пример: нетипизированный файл

7

```
const  
  RecSize = 13;
```

```
var  
  PersonFile: file;
```

0-я запись													1-я запись									
1	2	3	4	5	6	7	8	9	10	11	12	13	1	2	3	4	5	6	7	8	9	10
#6	И	в	а	н	о	в	М	#20	#3	Р	У	С	#8	С	и	д	о	р	о	в	а	Ж

Текстовые файлы

8

□ *Текстовый файл*

- Структура, стандартная для текстовых файлов ОС
- Записи текстового файла – символьные строки переменной длины
- Строки разделены комбинацией символов «*конец строки*».
 - «перевод каретки» (#13)
 - «перевод строки» (#10, может отсутствовать).
- Файл завершается символом «*конец файла*» (#26).
- <Имя переменной>: **Text**

Пример: текстовый файл

9

```
var  
  PersonFile : Text;
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1-я строка	И	в	а	н	о	в	#32	М	#32	2	0	#13	#10		
2-я строка	С	и	д	о	р	о	в	а	#32	Ж	#32	1	9	#13	#10
...															
п-я строка	Я	к	о	в	л	е	в	#32	М	#32	4	0	#26		

Основные операции с файлами

10

- `Assign(<Файл>, <Имя>)`
Назначение файловой переменной дискового файла с указанным именем.
- `Reset(<Файл>)`
Открытие файла для чтения.
- `Rewrite(<Файл>)`
Открытие для записи с предварительной очисткой.
- `Append(<Файл>)`
Открытие для записи без предварительной очистки (только для текстовых файлов). Новые записи будут добавляться в конец файла.

Основные операции с файлами

11

- ❑ **Close(<Файл>)**
Заккрытие файла.
- ❑ **Write([<Файл>,) <Список вывода>)**
Вывод значений из списка вывода в файл. Если файловая переменная не указана, то вывод происходит в стандартный файл вывода Output (на экран).
- ❑ **Read([<Файл>,) <Список ввода>)**
Ввод значений из файла в переменные, указанные в списке ввода. Если файловая переменная не указана, то ввод происходит из стандартного файла ввода Input (с клавиатуры).

Основные операции с файлами

12

□ WriteLn, ReadLn

Определены только для текстовых файлов.

Аналогичны `Write` и `Read`, но после операции ввода-вывода производят смену строки в файле.

□ Eoln(<Файл>)

Выдает `True`, если достигнут конец строки (только для текстовых файлов).

□ Eof(<Файл>)

Выдает `True`, если достигнут конец файла.

Пример 1: работа с текстовым файлом

13

```
{ Создание копии текстового
  файла с нумерованными
  строками. }
var
  Inp, Out: Text;
  S: String;
  L: LongInt;
const
  InpFName='input.txt';
  OutFName='output.txt'
```

```
begin
  Assign(Inp, InpFName);
  Reset(Inp);
  Assign(Out, OutFName);
  Rewrite(Out);
  L := 0;
  while not Eof(Inp) do
  begin
    ReadLn(Inp, S);
    L := L + 1;
    WriteLn(Out, L:4, S);
  end;
  Close(Inp);
  Close(Out);
end.
```

Пример 2: работа с текстовым файлом

14

```
{ Подсчет количества символов в
заданном файле.}
var
  F: Text;
  S, Name: String;
  Sum: LongInt;

begin
  Write('Введите имя файла =>_');
  ReadLn(Name);
  Assign(F, Name);
  Reset(F);
```

```
Sum := 0;
while not Eof(F) do
begin
  ReadLn(F, S);
  Sum:=Sum+Length(S);
end;
Close(F);
WriteLn(Name,
  ' содержит ', Sum,
  ' символов.');
```

```
end.
```

Операции с типизированными файлами

15

- `Seek(<Файл>, <номер>)`
Установка записи файла с указанным номером как текущей (нумерация с 0).
- `FilePos(<Файл>): LongInt`
Номер текущей записи файла (нумерация с 0).
- `FileSize(<Файл>): LongInt`
Количество записей в файле.
- `Truncate(<Файл>)`
Удаление всех записей файла после текущей.

Пример: работа с файлом записей

16

```
{ Создать файл записей
и выдать записи в
обратном порядке. }
const
  N=10;
  FName='person.rec';
type
  Person=record
    Name: String[10];
    Gender: Char;
  end;
var
  F: file of Person;
  P: Person;
  i: LongInt;
```

```
begin
  Assign(F, FName);
  Reset(F);
  for i:=1 to N do begin
    Write('Введите ФИО и пол: ');
    ReadLn(P.Name, P.Gender);
    Write(F, P);
  end;
  Close(F);
  Reset(F);
  for i:=1 to N do begin
    Seek(F, FileSize(F)-i);
    Read(F, P);
    WriteLn(P.Name, ' ', P.Gender);
  end;
  Close(F);
end.
```


Операции с нетипизированными файлами

17

- Rewrite/Reset(<Файл>, <Размер записи>)
Создать/открыть для чтения/записи нетипизированный файл с указанным размером записей.
- BlockRead(<Файл>, <Буфер>, <Читать записей> [, <Реально считано записей>])
Чтение указанного количества записей из файла в буфер. Может выдавать количество реально считанных записей.
- BlockWrite(<Файл>, <Буфер>, <Писать записей> [, <Реально записано записей>])
Запись указанного количества записей из буфера в файл. Может выдавать количество реально записанных записей.

Пример: работа с нетипизированным файлом

18

```
{ Выдать содержимое  
исполняемого файла на  
экран. }  
Program PrintSelfEXE;  
const  
  Len=80;  
var  
  F: file;  
  S: String[Len];  
  i,Res: LongInt;
```

```
begin  
  Assign(F, ParamStr(0));  
  Reset(F,Len);  
  repeat  
    BlockRead(F,S,1,Res);  
    for i:=1 to Res do  
      Write(Len[i]);  
      WriteLn;  
  until res<>Len;  
  Close(F);  
end.
```

Заключение

- Файл – именованная совокупность взаимосвязанных данных, хранящихся на внешних носителях. Файлы состоят из компонент, называемых записями.
- Виды файлов: типизированные, текстовые, нетипизированные файлы.
- Основные операции с файлами: Assign, Reset, Rewrite, Append, Read, Write, Close, Eoln, Eof.